CLAIMS

What is claimed is:

1. A method, comprising:

    identifying a candidate representing a plurality of instructions of a plurality of threads

    that perform one or more external memory accesses, the external memory

    accesses having a substantially identical base address; and

    inserting at least one of directives and instructions into an instruction stream

    corresponding to the identified candidate to maintain contents of at least one

    of a content addressable memory (CAM) and local memory (LM) of a

    processor and to modify at least one of the external memory access to access

    at least one of the CAM and LM of the processor without having to perform

    the respective external memory access.

2. The method of claim 1, further comprising:

    partitioning the plurality of instructions of the external memory accesses into one or

    more sets of potential candidates based on dependency relationships of the

    instructions; and

    selecting one of the potential candidate sets as the candidate, instructions of the

    candidate satisfying a predetermined dependency relationship.

3. The method of claim 2, further comprising converting addresses of each external

    memory accesses into a form having a base address and an offset.

4. The method of claim 3, wherein the base address is a non-constant part and the offset is

    a constant part of the converted address.

5. The method of claim 3, further comprising screening out one or more ineligible candidates from the potential candidates, wherein the ineligible candidates include a base address that is different from a remainder of the potential candidates.

6. The method of claim 3, further comprising grouping multiple potential candidates having substantially identical base address into a single candidate, wherein a group having most of the potential candidates is selected as a final candidate for caching.

7. The method of claim 1, wherein the identifying the candidate further comprises:

   performing a copy-forward transformation on addresses of each of the external memory accesses; and

   performing at least one of a global value numbering operation and a constant folding operation for each thread.

8. The method of claim 3, further comprising:

   for each thread, reserving a sufficient space in the local memory to store data portions of cache lines; and

   inserting a caching instruction prior to each of the external memory accesses.

9. The method of claim 8, further comprising seeking the base address of each external memory access in the CAM to determine whether the CAM includes an entry that contains the base address being sought.

10. The method of claim 9, wherein if the CAM includes an entry containing the base address being sought, the method further comprises:

    determining an offset of the local memory based on the entry of the CAM containing the base address being sought; and

    accessing data from an entry of the local memory referenced by the determined offset.

11. The method of claim 9, wherein if the CAM does not includes an entry containing the base address being sought, the method further comprises allocating a least recently used (LRU) entry of the CAM having a base address of a previous external memory access.

12. The method of claim 11, further comprising:

loading data of a current external memory access from the external memory into an entry of the local memory referenced by the allocated LRU entry; and

storing the base address of the current external memory access in the LRU entry of the CAM replacing the base address of the previous external memory access.

5

13. The method of claim 11, further comprising:

examining the base address of the previous external memory access in the allocated LRU entry to determine whether the base address is valid; and

replicating data of an entry in the local memory corresponding to the allocated LRU entry to a location of the external memory based address of the previous

10 external memory access.

14. A machine-readable medium having executable code to cause a machine to perform a method, the method comprising:

identifying a candidate representing a plurality of instructions of a plurality of threads that perform one or more external memory accesses, the external memory accesses having a substantially identical base address; and

15 inserting at least one of directives and instructions into an instruction stream corresponding to the identified candidate to maintain contents of at least one of a content addressable memory (CAM) and local memory (LM) of a processor and to modify at least one of the external memory access to access at least one of the CAM and LM of the processor without having to perform

20 the respective external memory access.

15. The machine-readable medium of claim 14, wherein the method further comprises:

    partitioning the plurality of instructions of the external memory accesses into one or more sets of potential candidates based on dependency relationships of the instructions; and

    selecting one of the potential candidate sets as the candidate, instructions of the candidate satisfying a predetermined dependency relationship.

5

16. The machine-readable medium of claim 15, wherein the method further comprises converting addresses of each external memory accesses into a form having a base address and an offset.

17. The machine-readable medium of claim 15, wherein the method further comprises screening out one or more ineligible candidates from the potential candidates, wherein the ineligible candidates include a base address that is different from a remainder of the potential candidates.

18. The machine-readable medium of claim 15, wherein the method further comprises grouping multiple potential candidates having substantially identical base address into a single candidate, wherein a group having most of the potential candidates is selected as a final candidate for caching.

19. The machine-readable medium of claim 15, wherein the method further comprises:

    for each thread, reserving a sufficient space in the local memory to store data portions of cache lines; and

    inserting a caching instruction prior to each of the external memory accesses.

10

20. The machine-readable medium of claim 19, wherein the method further comprises seeking the base address of each external memory access in the CAM to determine whether the CAM includes an entry that contains the base address being sought.

21. The machine-readable medium of claim 20 wherein if the CAM includes an entry containing the base address being sought, the method further comprises:

determining an offset of the local memory based on the entry of the CAM containing the base address being sought; and

accessing data from an entry of the local memory referenced by the determined offset.

5

22. The machine-readable medium of claim 20, wherein if the CAM does not includes an entry containing the base address being sought, the method further comprises allocating a least recently used (LRU) entry of the CAM having a base address of a previous external memory access.

23. The machine-readable medium of claim 22, wherein the method further comprises:

loading data of a current external memory access from the external memory into an entry of the local memory referenced by the allocated LRU entry; and

storing the base address of the current external memory access in the LRU entry of the CAM replacing the base address of the previous external memory access.

10

24. The machine-readable medium of claim 22, wherein the method further comprises:

examining the base address of the previous external memory access in the allocated LRU entry to determine whether the base address is valid; and

replicating data of an entry in the local memory corresponding to the allocated LRU entry to a location of the external memory based address of the previous

15 external memory access.

25. A processor, comprising:

a plurality of microengines having a content addressable memory (CAM) and a local memory respectively to perform a plurality of threads substantially concurrently, each of the plurality of threads including one or more

instructions performing at least one external memory access based on a base address that is substantially identical,

wherein the base address is examined in the CAM to determine whether the CAM includes an entry containing the base address, and

5   wherein an entry of the local memory corresponding to the entry of the CAM is accessed without having to accessing the external memory, if the CAM includes the entry containing the base address.

26.   The processor of claim 25, wherein the CAM of the microengines comprises comprises a least recently used (LRU) logic to allocate an LRU entry of the CAM linking with an entry of the local memory, wherein the allocated LRU entry is used to cache the external memory access for subsequent accesses to an identical location of the external memory.

27.   The data processing system of claim 26, wherein the LM comprises an indexing logic to provide an index pointing to an entry of the LM based on a reference supplied by the LRU logic.

28.   A data processing system, comprising:

a processor;

10   a memory coupled to the processor; and

a program instruction, when executed from the memory, causes the processor to

identify a candidate representing a plurality of instructions of a plurality of threads that perform one or more external memory accesses, the external memory accesses having a substantially identical base

15   address, and

insert at least one of directives and instructions into an instruction stream corresponding to the identified candidate to maintain contents of at least one of a content addressable memory (CAM) and local memory

(LM) of an executing processor executing the plurality of threads and to modify at least one of the external memory access to access at least one of the CAM and LM of the executing processor without having to perform the respective external memory access.

5

29. The data processing system of claim 27, wherein the plurality of threads is executed by a plurality of microengines of the executing processor respectively, and wherein each of the microengines of the executing processor includes a CAM and a LM.

30. The data processing system of claim 28, wherein the CAM of the microengines comprises a least recently used (LRU) logic to allocate an LRU entry of the CAM to cache a current external memory access for subsequent identical external memory access, if the CAM does not contain the base address of the current external memory access, and wherein the LM comprises an indexing logic to provide an index pointing to an entry of the LM based on a reference supplied by the LRU logic.